PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 4 : G06F 9/46, 13/2

A1

(11) International Publication Number: WO 88/ 03682

(43) International Publication Date: 19 May 1988 (19.05.88)

(21) International Application Number: PCT/US87/02812

(22) International Filing Date: 29 October 1987 (29.10.87)

(31) Priority Application Numbers: 926,568
926,588
926,567
926,738

(32) Priority Dates: 4 November 1986 (04.11.86)
4 November 1986 (04.11.86)
4 November 1986 (04.11.86)
4 November 1986 (04.11.86)

(33) Priority Country: US

(71) Applicant: UNISYS CORPORATION [US/US]; P.O. Box 500, Blue Bell, PA 19424 (US).

(72) Inventors: PEACOCK, Richard, Browning ; 400 Westgate Village, Frazer, PA 19355 (US). MURPHY, Philip, Arthur ; 296 Anthony Road, King of Prussia, PA 19406 (US). MISSIMER, David, Ross ; William Henry Apts., Norwood 313, Malvern, PA 19355 (US).

(74) Agent: STARR, Mark, T.; Unisys Corporation, P.O. Box 500, Blue Bell, PA 19424-0001 (US).

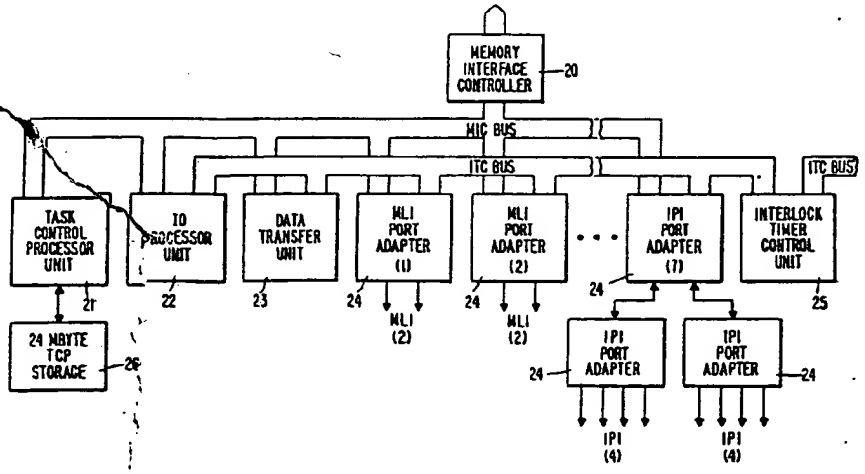
(81) Designated States: AT (European patent), BE (European patent), CH (European patent), DE (European patent), FR (European patent), GB (European patent), IT (European patent), JP, LU (European patent), NL (European patent), SE (European patent).

Published
With international search report.
Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: I/O SYSTEM FOR OFF-LOADING OPERATING SYSTEM FUNCTIONS

(57) Abstract

An I/O processor (22) and memory (12) where a number of queues or linked control blocks (IOCB 1 and IOCB 2) are maintained for each device connected to the I/O processor (22), there is a control block for every operation to be performed by a particular device. A device may be an I/O bus (13a), a controller unit (14) or a peripheral unit (15), the I/O processor (22) maintains a table (46) of different combinations of buses and peripheral controllers that may be used to access a given peripheral unit (15) and selects that combination with the least frequency of use. A portion of main memory (12) is assigned as a single cache so that when the I/O processor (22) accesses a data segment in one of many disk drives (15), the entire disk drive track (59a, b and c) being accessed is read into the assigned cache portion of main memory since following data requests would most likely be made therefrom. The I/O system (13) is provided with a Task Control Processor (21) which provides for the scheduling of the different central processors (10) for the highest priority processes to be run. When an I/O initiate I/O operation is detected, the respective processor (10) is released from the process that it is currently running and can be assigned to the next highest priority process. When requested I/O operation has been completed, the Task Control Processor (13) is signalled so that the Task Control Processor (13) can put the requesting process back into the priority list of processes to be run by the main central processors (10).



FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	FR	France	ML	Mali
AU	Australia	GA	Gabon	MR	Mauritania
BB	Barbados	GB	United Kingdom	MW	Malawi
BE	Belgium	HU	Hungary	NL	Netherlands
BG	Bulgaria	IT	Italy	NO	Norway
BJ	Benin	JP	Japan	RO	Romania
BR	Brazil	KP	Democratic People's Republic of Korea	SD	Sudan
CF	Central African Republic			SE	Sweden
CG	Congo	KR	Republic of Korea	SN	Senegal
CH	Switzerland	LI	Liechtenstein	SU	Soviet Union
CM	Cameroon	LK	Sri Lanka	TD	Chad
DE	Germany, Federal Republic of	LU	Luxembourg	TG	Togo
DK	Denmark	MC	Monaco	US	United States of America
FI	Finland	MG	Madagascar		

I/O SYSTEM FOR OFF-LOADING OPERATING SYSTEM FUNCTIONS

Field of the Invention

This invention relates to an input/output system for a very large computer system and more particularly to such an I/O system wherein the I/O software functions are implemented in the I/O hardware system.

Description of the Prior Art

A very large multi-processing system or a very large single processing system adapted for multi-programming require large amounts of data in their various computations and thus are provided with a hierarchy of storage units ranging from main memory to bulk storage devices such as disk drives to peripheral devices such as tape drives, and the like. The system is provided with I/O controllers which control the data transfer from the peripheral devices to the disk storages or from the peripheral devices and the disk storages to main memory. However, in such prior art systems the central processors are required to decode the I/O instructions and send the respective control signals to the I/O controllers and this takes up an unduly amount of the processors's execution time. Examples of such prior art controllers are disclosed in the Annunziata et al. U. S. Patent No. 3,432,813 and the Calta et al. U. S. Patent No. 3,447,138.

Attempts have been made to free the central processor from this I/O execution so that the central processor can spend more time on user jobs by supplying a separate general purpose processor to operate independently in the control of input/output data transfers. However, there must be some communication between the two processors in order to assure that the data required by the main central processor is received in its main memory prior to the central processor utilizing that data.

Input/output operations include more than just data transfers between the periphery and main memory. For example, the I/O system must control such non-data transfer operations as rewinding tape reels and the like. Furthermore, in very large data processing systems, there are

a number of different buses and peripheral controllers that may be chosen to optimize through-put from the periphery to main memory and vice versa. In addition, particularly in regard to disk drives, a cache memory is provided to store the most recently accessed data segments in that cache, which data segments are more likely to be accessed again shortly. However, these disk functions are handled by an operating system running on a central processor, requiring the processor to spend additional time that could be employed in running user jobs or tasks.

Statistical studies indicate that a major portion of each processor's time, in a multi-processing system, is employed in executing operating system functions. From these studies, it is estimated that the overhead of such management functions has been anywhere between 10 percent and 50 percent, and occasionally even higher. Furthermore, a goodly portion of the time that the corresponding central processor is executing operating system functions is employed in establishing process priority, performing functions on events (to be defined below) and initiating input/output operations. If these functions could be removed from the operating systems, then the through-put of the data processing system should be substantially enhanced.

INTENTIONALLY
BLANK

BRIEF DESCRIPTION OF THE DRAWINGS

The above and other objects, advantages and features of the present invention will become more readily apparent from a review of the following specification when taken in conjunction with the drawings wherein:

FIG. 1 is a block diagram of a system employing the present invention;

FIG. 2 is a block diagram of the input/output system of the present invention;

FIG. 3 is a diagram between the relation of the various tables employed by the present invention which tables reside in both the I/O Processor, main memory and in port adapters;

FIG. 4 is a schematic diagram of the I/O Processor of FIG. 2;

FIGS. 5A-D represent other tables in memory employed by the present invention and the relation therebetween;

FIG. 6 is a schematic diagram of the Task Control Processor of Fig. 2; and

FIG. 7 is a diagram of a portion of a disk drive as employed with the present invention.

GENERAL DESCRIPTION OF THE PREFERRED EMBODIMENT

A system employing the present invention is illustrated in FIG. 1. As shown therein, this system is a very large multi-processing system having a plurality of central processors 10 which are coupled to another plurality

of main memory modules 12 by way of memory controller 11 which allows any processor to address any memory module.

More specifically, the present invention resides in I/O system 13 which controls all communication and data transfer between peripheral units 15 and main memory modules 12. As will be discussed further, I/O 13 can also communicate with respective central processors 10 by way of memory controller 11. It is to be noted in FIG. 1, that there are a plurality of controllers 14 coupled between respective peripheral units 15 and I/O system 13 by way of a plurality of different buses 13a. That is to say, that a given peripheral unit 15 can be accessed by I/O system 13 by way of alternative combinations of buses 13a and controllers 14. Peripheral units 15 may include any type of peripheral device or storage including large disk drives in which are stored the operating systems of the data processing system of FIG. 1 and also critical user data.

I/O system 13 of FIG. 1 is shown in more detail in FIG. 2 and contains a number of different units that interface by way of memory interface controller 20 with memory controller 11 of FIG. 1. As shown in FIG. 2, I/O system 13 includes Task Control Processor 21 which handles all process scheduling on respective central processors 10 of FIG. 1 and also keeps track of various events upon which different processes might be waiting. I/O processor 22 is the heart of the present invention and performs all the functions that have been referred to above and will be more fully described below. Data transfer unit 23 is employed to move data between different areas of memory to other areas of memory and is specifically useful in the disk cache mechanism of the present invention. Port adapters 24 are basically bus drivers for the respective buses 13a of FIG. 1 although they may employ different protocols. Interlock timer control 25

distributes interrupts to the various buses and also provides a queue locking mechanism by which it is guaranteed that shared queues (PQ, BQ) are not corrupted by multiple simultaneous access. Task Control Processor 21 is described in detail in the Jennings et al. application U. S. Serial Number 787,781, filed October 15, 1985 and assigned to the same assignee as the present invention.

As has been indicated above, the function of the present invention is to relieve the operating systems and the respective central processor 10, which execute those operating systems, of all I/O operations so that central processors 10 will have more time for the execution of user jobs. When a given central processor 10 is executing a process from one of memory modules 12 and encounters an I/O operation, the corresponding I/O control block is created and the I/O instruction is sent to I/O system 13 by way of memory controller 11 and the processor 10 is released to begin executing the next highest order process from one of memory modules 12. When the I/O operation has been completed, the requesting process is then rescheduled in a priority list of processes for further execution by the next available central processor 10.

Before describing the details of I/O processor 22 of FIG. 2, a description will first be given of the data structure linkages or linkages between tables employed by the I/O processor which are illustrated in FIG. 3. The I/O start instruction or ASYNC-SCHEDULE command is received by input message buffer 30 of FIGS. 3 and 4. In FIG. 3, there is only one such command which consists of four words. The first word contains an operation command and also an indication of the initiating process which in the system of the present invention and in the above-described Jennings et al. application are also referred to as a stack. The second word

of the message or instruction contains a device number which identifies the device to be employed by the I/O operation. The third word contains a reference to an I/O control block (IOCB) which is created by a central processor and stored in main memory as will be more fully described below. The fourth word contains the time of the command initiation.

The I/O processor then takes the device number which addresses device table 45 of FIGS. 3 and 4 which contains an entry for each device in the system which entry includes the current status of that device and if the device is not available or not in an idle state, then the device control block is stored in a device queue in main memory until the device is idle. As employed in the present application, the term "device" is employed to mean either a bus 13a of FIG. 1, a controller 14 of FIG. 1, or a peripheral unit 15. If the device is idle, then a reference is obtained to path group table 46 of FIGS. 3 and 4 which specifies which controllers and associated buses are to be employed to access the device which is normally a peripheral unit. In FIG. 3, the entry in path group table 46 indicates that three controllers can be used. Controllers servicing a unit have equal priority, and the IOP attempts to keep the load seen by each controller well balanced by its path selection algorithm. Buses servicing a controller are listed in the path group entry for that controller in priority order. Once the bus and controller combination for a given device (usually a unit) is determined, reference is made to interlock translation table 25a of interlock timer control unit 25 of FIG. 2. The proper path queue is locked via ITC. The control block is enqueued into the path queue. The path queue is unlocked and the IO bus is interrupted again via the ITC.

When the device was selected, reference was made back to the indirect reference queue for that unit in memory and the contents of unit queue 60 of FIG. 3 just show that indirect reference or Q header. This in turn allows the I/O processor to fetch the appropriate control blocks from main memory which are passed to the I/O processor which could not be executed immediately. Similarly, after the path group has been selected, reference is made to path queue 62 which again contains a Q header pointing to the control blocks for the selected bus or controller. These control blocks or parts of them are passed on to the selected controller and so forth until the I/O operation is complete, in which case they are passed back to memory. When the I/O operation has been finished, the control blocks are passed back to main memory and task control processor 21 of FIG. 2 reschedules the requesting process.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

A block diagram of the I/O processor of the present invention is illustrated in FIG. 4. It is to be noted therein that this processor is controlled by three different control sequencers or control stores 31, 41, and 51, the purpose of which are to control different portions of the I/O processor in a concurrent manner. Among the advantages of this, I/O processor of FIG. 4 can send a message acknowledgment signal to a requesting central processor of FIG. 1 before the processing of that message is actually begun and thus speeds up the release of the requesting central processor so that it may be assigned to other user tasks.

Memory control sequencer 51 controls the main input bus or memory read bus by which messages are transferred to message and input data buffer 30 by way of read registers 30a

and 30b. In addition, primary control sequencer 31 controls main arithmetic logic unit 33 which is used to calculate device, unit and other addresses according to the particular command being implemented. Inputs to ALU 33 are by way of B register 34 and accumulator 35 which in turn are supplied by multiplexer 36 and multiplexer and rotator 37. The output of ALU 33 is used to update results in local memory; to update the rotator for field manipulation; to drive the interlock interface; and to send parameters to the auxiliary control sequencer via its B register. The output from ALU 33 is sent to multiplexer and rotator 37 and hence to accumulator 35 either for further manipulation or for transmission to interlock interface 39 or to save results in local memory by way of multiplexer 38. Multiplexer 38 may also receive information from B register 34 when that information is to be sent to interlock interface 39.

Referring back to FIG. 2, the various units therein are connected by two buses, namely memory interface controller bus MIC which communicates with main memory by way of memory controller 11 of FIG. 1 and also interlock timer control bus ITC which communicates with, among other things, port adaptors 24 and interlock timer control unit 25. In FIG. 4, I/O processor communicates with MIC bus by way of memory read bus and message and input data buffer 30 and also memory write bus by way of memory output register 56. The I/O processor of FIG. 4 communicates with the interlock timer control bus ITC by way of interlock interface 39.

Continuing on with the discussion of FIG. 4, the I/O processor shown therein is provided with an auxiliary arithmetic logic unit 43 and appropriate input and output registers and multiplexers for off-loading of certain tasks by primary control sequencer or control store 31 to auxiliary

ALU 43 and associated registers which are under control of auxiliary control sequencer or control store 41. (not labelled) To transfer tasks from primary control store 31 to auxiliary control store 41, primary control store 31 addresses auxiliary control store 41 by inserting operation codes and associated parameters into spare areas of the path group table memory, and setting the auxiliary sequencer interrupt. When the auxiliary sequencer has completed the requested function, it writes its results into a spare area of the path group table memory, and resets the auxiliary sequencer interrupt. The primary sequencer then reads the results directly. Auxiliary control store 41 and associated logic is employed to complete the task of determining the status of a selected device stored in device table 45 and also to select the appropriate bus controller pair as selected by addressing path group table 46. It is also employed in the arithmetic calculations required for disk caching.

The mechanisms of I/O processor of FIG. 4 described thus far have dealt primarily with handling the receipt of a start I/O instruction or more specifically an ASYNC-SCHEDULE instruction and the selection of the appropriate device or unit and corresponding bus-controller combination for the subsequent transfer of an I/O control block from main memory to the selected controller and port adaptor which drives the bus.

To handle information transfer from I/O processor of FIG. 4, memory control sequencer or control store 51 is provided to control this transfer concurrently with primary control store 31 but in synchronization therewith since both control stores 31 and 51 access local memory 52. This synchronization is accomplished through two separate shared

hardware queue structures within local memory 52. The local memory is designed to run at twice the frequency of the surrounding logic, so that during each machine cycle, the memory control sequencer can read a given location and the primary control sequencer can either read or write a given location. The PCS makes requests of the MCS by writing entries into the queue areas (fetch Q or store/MSG Q). The MCS services the requests by reading the queues and accessing the data in LM.

In FIG. 4, data and information transfer from the port adapters and other units of FIG. 2 is received from interlock interface 39 by way of local memory 52 which serves a variety of purposes. Information transfer to main memory is by way of multiplexer 53, staging register 54, command modification unit 55 and memory output register 56. The modified commands generated by command modifier unit 55 are then returned back by way of command update function unit 57.

Reference has been made above to the storing of device control block messages in a queue in memory when the device being requested is not idle or otherwise not available, to queue headers and to I/O control blocks (I/OCB). These will now be more fully described in relation to FIGS. 5A-D. These various information structures represent structures stored in a portion of main memory set aside as an I/O work space which is reserved for the system of the present invention.

FIG. 5A is a diagram of a table in main memory of device sections, one for each device in the system where, as was mentioned above, the device may be one of the I/O buses 13a of FIG. 1 (or more particularly one bus of the port adapters 24 of FIG. 2 which drive the corresponding buses), one of controllers 14 of FIG. 1 or one of peripheral units 15

of FIG. 1. As indicated in FIG. 5A, there may be up to 128 I/O buses, up to 512 (less the number of I/O buses) controllers with the remaining devices being peripheral units up to 4,096 less the number of I/O buses and controllers.

Each device section includes two queue headers which, as more thoroughly described below, contain pointers or memory addresses to different I/O control blocks (IOCB) there being one such block for each operation currently being performed by a device in the system. An exception to this is that the unit device sections only contain one such queue header with the other portion of that section being used as a scratch pad area.

The general format of each queue header is illustrated in FIG. 5B and the unit scratch area format is illustrated in FIG. 5C. In FIG. 5B, the queue header is made up of four control words. The first word contains control and status information. The second word is a pointer or an address to main memory at the beginning of the first I/O control block for that particular device. The third word is a pointer or address to main memory of the last or most recent I/O control block so that these two pointers indicate the beginning and the end of a particular queue of linked together control blocks for different jobs or tasks that have been requested of the device in question. This will be more thoroughly described in relation to FIG. 5D.

Finishing the description of FIG. 5A, the I/O bus device areas include a bus queue header into which operators for controlling the corresponding I/O bus are enqueued. These device areas also include a queue header for controlling the transfer of the results of the device operation via the corresponding port adapter. Each of the controller device sections includes a path queue header for passing the corresponding control block to the respective

controller that was selected in the bus-control combination as was described above and a unit queue header which points to control blocks pending for the controller itself. Such path queue headers and unit queue headers were described above in relation to FIG. 3.

Turning now to FIG. 5D, the queuing mechanism of the present invention for linking together of control blocks for different jobs or tasks requested on the particular device will now be described. As was indicated above, most of the requests for devices are requests for peripheral units 15 of FIG. 1 and there is an I/O control block for each job or task that has been requested of the respective devices.

When an input message comes into input message buffer 30 of FIGS. 3 and 4, it specifies the device number and also a reference or memory address to the particular control block for the job to be performed on that device. If the device is idle, the control block is fetched; path selection is performed; the IOCB is linked into the selected controller's path; and the I/O bus servicing that controller is interrupted via the ITC mechanism already described. If the device is not available, then the control block reference will be inserted as a tail pointer in the particular device queue header as indicated in FIG. 5D. It will also be inserted as the head pointer if the unit queue was previously empty. A field in the unit queue control word indicates the device is idle. As additional particular requests for that device come in, then the second requested control block address is inserted as the next link memory address in the head I/O control block and also in the tail pointer of the particular device queue header, as illustrated in FIG. 5D. In this manner, many requests for a particular device can be queued with the queue mechanism of the present invention.

Sometimes the various I/O control blocks, such as those represented in FIG. 5D, are employed by the I/O processor of FIG. 4 to communicate commands to the various devices to essentially cause transitions of the device. That is to say, a device can be in one of four states: free, saved, ready, or suspended. Examples of device management procedures implemented by the processor of FIG. 4 include: ACQUIRE which, if successful, transitions a device from a free state to a saved state; READY, which if successful, transitions a device from saved to ready; SUSPEND which, if successful, transitions a device from ready to suspended; SAVE which, if successful, transitions a device from ready to saved; and FREE which, if successful, transitions a device from a saved state to a free state. These procedures can return to the operating system with an error for various reasons, including that the specified device was not in the proper initial state. The state of every device is maintained in the device table 45 of FIG. 4 within the IOP.

Referring back to FIG. 5C, certain functions that can be performed by the mechanism of the present invention as thus disclosed will now be described. One of these features is that of disk mirroring which is a technique in which more than one copy of critical data is kept on separate disk devices so, that even in the case of serious disk failure, the system will still be kept running. The I/O processor of FIG. 4 supports disk mirroring in two main ways. First, for a disk read to a mirrored set, it will select the best unit from among the mirrored set to which the request is to be forwarded. The major components of delay in completing disk access are: seek time, which is the time that the disk arm is moving to the proper cylinder or track; rotational latency, which is the time after the seek completed but before the data is actually under the read head of the disk;

and data transfer, which is the time required to extract the requested data from the media. For those readers not familiar with disk mechanisms, a section of such a disk 59 is illustrated in FIG. 7 showing the individual tracks A, B, C.... The processor of FIG. 4 attempts to minimize the seek time for selecting a read unit from the mirrored set. The second way that the I/O processor supports disk mirroring is that on disk writes, the processor synchorizes all of the individual disk writes forming a single logical write, accumulating times and results into one and sending only one response back to the initiating user. To this end, the second and third words of the unit scratch area of FIG. 5C are employed.

Another feature implemented by the mechanism of the present invention is that of disk caching. In order to reduce the amount of time which I/O processes take, the processor of FIG. 4 implements a hardware managed disk cache in main memory. The only type of peripheral whose performance is critical to the overall system performance is the disk unit in which is stored the operating system, user data bases, and other frequently accessed information. The processor of FIG. 4 is allocated a very large area of system memory which it uses to keep copies of recently accessed disk tracks in the hope that further accesses will be made to the same track which is more often than not the case at hand. The advantage of this is that different access requests to a particular track on the disk unduly tie up the corresponding buses and controllers used to access the corresponding disk. Also I/O's which are disk cache hits can complete about three orders of magnitude faster than the physical disk accesses. To this end, the fourth word in the unit's scratch area of FIG. 5C is employed by the processor of FIG. 4.

As is described above in relation to FIG. 2, the I/O system of the present invention includes task control processor 21, which handles all process scheduling of the respective central processors 10 of FIG. 1 and also keeps track of various events upon which different processes might be waiting, including I/O operations. Thus, there is a certain cooperation between I/O processor 22 of FIG. 2 and task control processor 21 since some of the events upon which user processes may be waiting include I/O operations.

A functional diagram of task control processor 21 is illustrated in FIG. 6. The two principal functional elements shown therein are process table 61 and event table 60a. Process table 61 and process statistics table 60b contain the information as to the status of all tasks or processes scheduled to be run on the system of FIG. 1. In the described embodiment of the present invention, there can be 4 K such tasks or processes running on the system at any one point in time.

The status information of the processes in process table 61 are arranged as a queue or a linked list of processes according to the priority of the processes involved. As was indicated above, such a task control processor is described in detail in the above referenced Jennings et al. U. S. Patent Application 787,781.

As used in the remaining portion of this application, the terms "task", "process", and "stack" are used as being synonymous where a stack is a natural physical location in main memory and the respective task or processes are independent of one another and occupy the corresponding stack space. Thus, the terms "stack number", "task number", and "process number" are used synonymously and are the actual addresses to process table 61 of FIG. 6 of the corresponding process status information.

Event table 60a is employed to contain information as to the status of various event designations (in the present application, I/O operations) called for by user processes running on the system. In the embodiment of FIG. 6, there may be a maximum of 512 K such events being utilized at any one time. When a process being executed by a particular processor 10 of FIG. 1 requires an event designation, it requests the allocation of such a designation from the task control processor of FIG. 6 which then allocates an unallocated event designation to that process and sends an event token to be placed in main memory on top of the particular stack whose process requested the event designation. Event table 60a then upgrades the event information to indicate that the event has been allocated. The event token is made up of the event address to event table 60a and also certain coded bits to ensure that one of the processors 10 of FIG. 1 does not accidentally create its own event token. Event table 60a is also employed to maintain a linked list of various processes requesting a particular event that has already been allocated and assigns that event to the highest priority process requesting that event when the event is freed or liberated by its owning process.

An event designation does not specify the particular function for which the event was allocated. This is done by the requesting process. Event table 60a serves the purpose of maintaining the status of the event, e.g., whether it is available for allocation, whether it has occurred, what processes are waiting on it, etc.

Continuing on with the description of FIG. 6, support logic 62 is employed to insert information fields into event table 60a, statistics table 60b and link table 60c as well as to extract fields therefrom as required. Local

memory 63 serves as an output buffer and also maintains a processor table which indicates which processes are currently running on the respective processors 10 of FIG. 1.

Message transmission to the other processors of FIG. 1 are by way of memory controller 11 of FIG. 1 from output register 69 of FIG. 6. Messages are received from controller 11 by way of input register 65 to message buffer 64. As indicated in FIG. 6, the various functional units thus described have inputs to arithmetic logic unit module 66 by way of arithmetic logic unit input multiplexer 67. Arithmetic logic unit module 66 is employed to compute process priorities as described above and also to form messages for transmission to other processors of the system. All of the functional units of FIG. 6 are under the control of sequence control store 60 and are activated by the receipt of an external processor request by message buffer 24, where the request command is decoded by control store 60.

Task control processor 21 is designed to relieve the master control program of many of its most time consuming functions and, along with I/O processor 22 of FIG. 2, to relieve the master control program of most of the functions involved with I/O operations. Thus, in the present invention, when a processor is executing a particular user process, and encounters a reference to data not in main memory, it requests that task control processor 21 of FIG. 2 allocate an event token to that process and then initiates a procedure call for an I/O procedure for creating I/O start instruction or ASYNC/SCHEDULE command which is transferred to I/O processor 22 of FIG. 2 and message input data buffer 30 of FIG. 4. When the I/O process has been completed, primary control sequencer 31 creates a message which is sent from I/O processor 22 to task control processor 21. This in turn sets

a particular bit in the particular event location of event table 60a which results in the awakening of all processes which are currently waiting on the designated event so that they may be rescheduled by the task control processor for execution by the next available processor 10 of FIG. 1 according to their priority with no central processor involvement.

Brief mention was made above to disk caching by which each time a segment is fetched from a disk, its entire track is read into main memor as there is a higher probability that later I/O requests will shortly access that same track. By maintaining the disk cache in main memory, for all the disks in the system, conflicts between requests for various buses 13a of FIG. 1 and controllers 14 thereof are greatly reduced. It should be noted, that in a very large data processing system of the type employing the present invention, the operating systems will not only be quite large so as to require many disks for permanent storage, but so too will critical user data bases such as customer bank accounts and the like which may be very frequently accessed or updated. It would not be unusual in such a situation where all of the peripheral units 15 of FIG. 1 were disk drives except for some tape units for off-loading the system.

Because of the major storage requirements for I/O transfers, a goodly portion of main memory is assigned for such I/O operations. Thus, in FIG. 1, memory module 12a may be assigned as the I/O work area described above, memories 12b and 12c would be assigned as the disk cache while the remaining memory modules would be assigned to user programs and data.

Mention will now be made of data transfer unit 23 in FIG. 2 which is employed solely for the purpose of transferring data segments to and from the disk cache in memory module 12b and the user data base in the user portion of the memory. For reasons of manufacturing economics, this unit will be a circuit board which is the same as that which embodies the I/O processor 22 and illustrated in detail in FIG. 4, except, in FIG. 4, the auxiliary control sequencer and the units under its control as well as device table 45 and path group table 46 are not employed. However, this data transfer unit will contain different microcode sequences in its control stores than the I/O processor so as, in effect, to be a slave to the I/O processor which ultimately controls the entire I/O system 13 of FIG. 1.

In operation, when the ASYNC-SCHEDULE command is received by message and input data buffer 30 of FIG. 4 and it is determined that the device selected is a disk, primary control sequencer 31 will search the disk cache in main memory. If the operator is a READ, the primary control sequencer 31 of FIG. 4 in the I/O processor will, using the normal queuing structures, instruct its corresponding control store in the data transfer unit to transfer that data item from the disk cache in main memory to the user portion of the requesting process in main memory. If there is a miss during the search, then primary control sequencer 31 of FIG. 4 creates a new control block in its I/O work space for transfer to an appropriately selected port adaptor 24 to access the specific disk and its corresponding track and fetch the entire track back through the port adaptor to the disk cache portion of main memory. During this transfer, when the specifically selected data item has been stored in

the disk cache, I/O processor unit 22 of FIG. 2 signals data transfer unit 23 to fetch that item from the disk cache and transfer it to the user portion of the requesting process in memory.

A similar reverse process is employed for a disk write. When the ASYNC-SCHEDULE command is a disk write, the I/O processor will cause a search in the disk cache to see if the corresponding disk track is stored therein, and if it is, then the I/O processor signals data transfer unit 23 of FIG. 2 to transfer that data from the user portion of main memory to the appropriate disk track and also causes an appropriate port adaptor to update the particular disk from which that disk track was fetched. Otherwise, the disk write operation is performed like any other data transfer to a peripheral unit.

What is claimed is:

1. In a processing system having at least one central processor and a memory for storing a plurality of processes to be executed by said at least one central processor, which processes require different I/O events to occur before their execution can be completed, an input/output system comprising:

a task control processor having means coupled to said at least one central processor to receive a command to allocate an event token to a currently executing process and then to indicate that the currently executing process is in a wait state, said event token indicating that an I/O operation has been requested; and

an input/output processor coupled to said task control processor and to said at least one central processor to receive an input/output start command, said input/output processor being coupled to said task control processor to signal said task control processor that said input/output process has been completed.

2. An input/output system according to Claim 1 wherein said task control processor includes:

event table means coupled to said coupling means to store status information about various event tokens including whether that token has been allocated and whether that event has occurred.

3. An input/output system according to Claim 2 further including:

process table means for storing process designations of processes scheduled to be executed by said processing system including processes which are available to be executed and arranged according to assigned priorities.

4. An input/output system according to Claim 3 further including:

priority computation means coupled to said process table means to change the priority of various process designations in said process table means when new processes are scheduled for execution by said processing system.

5. An input/output system according to Claim 4 wherein:

said event table means contains a linked list of all event tokens that have been allocated to a particular process.

6. An input/output system according to Claim 5 wherein:

said event table means includes a link table means for storing a list of all processes having requested procurement of each of the particular event tokens.

7. An input/output system according to Claim 6 wherein:

said event table means includes a link table means for storing a list of all processes waiting on each of the allocated event tokens to have occurred.

8. In a processing system having at least one central processor and a memory for storing a plurality of processes to be executed by said at least one central processor, which processes require different I/O events to occur before their execution can be completed, an input/output system comprising:

a task control processor having means coupled to said at least one central processor to receive a command to allocate an event token to a currently executing process and then to indicate that the currently executing process is in a wait state, said event token indicating that an I/O operation has been requested; and

an input/output processor coupled to said task control processor and to said at least one central processor to receive an input/output start command, said input/output processor being coupled to said task control processor to signal said task control processor that said input/output process has been completed;

said task control processor including an event table means having a link table means for storing a list of all processes waiting on each of the allocated event tokens to have occurred.

9. An input/output system according to Claim 8 and including said task control processor wherein:

said event table means is coupled to said coupling means to store status information about various event tokens including whether that token has been allocated and whether that event has occurred.

10. An input/output system according to Claim 9 further including:

process table means coupled to said coupling means for storing process designations of processes scheduled to be executed by said processing system including processes which are available to be executed and arranged according to assigned priorities.

11. An input/output system according to Claim 10 further including:

priority computation means coupled to said process table means to change the priority of various process designations in said process table means when new processes are schedule for execution by said processing system.

12. A processing system having at least one central processor and a memory for storing a plurality of processes to be executed by said at least one central processor, which processes require different input/output events to occur, said system comprising:

an input/output processor means coupled to said memory;

a plurality of controller means;

a plurality of peripheral units; and

a plurality of buses coupling said peripheral units to various ones of said controller means and also coupling said controller means to said input/output processor means;

said input/output processor means including an input buffer to receive an input/output operation command specifying a particular unit and a reference to a control block in memory to control the operation of that unit, said input/output processor means also including a device table specifying the current status of activity of each of said units, said input/output processor means also including control means to address said memory to link the address therein of an input/output control block for a second request of a peripheral unit operation to the control block for a first peripheral unit operation, when said requested peripheral unit is already in operation.

13. A system according to Claim 12 wherein:

said memory contains a device table having a section for each device in the system, where a device may be a peripheral unit, a controller means, or a bus, each device section containing a location for an address to a control block for that device when that device is in operation.

14. A system according to Claim 13 wherein:
each device section of said device table includes a location for storing an address to a control block for the first device operation to be requested and also a storage location to contain an address to the control block for the last device operation to be requested.

15. A system according to Claim 14 wherein:
a portion of said device table sections are set aside for input/out bus operations.

16. A system according to Claim 15 wherein a portion of said device table is set aside for controller means operations.

17. A system according to Claim 16 wherein a portion of said device table is set aside for peripheral unit operations.

18. In combination, an input/output system and a memory, said system comprising:

an input/output processor means coupled to said memory;

a plurality of controller means;

a plurality of peripheral units; and

a plurality of buses coupling said peripheral units to various ones of said controller means and also coupling said controller means to said input/output processor means;

said input/output processor means including an input buffer to receive an input/output operation command specifying a particular unit and a reference to a control block in memory to control the operation of that unit, said input/output processor means also including a device table specifying the current status of activity of each of said units, said input/output processor means also including control means to address said memory to link the address therein of an input/output control block for a second request of a peripheral unit operation to the control block for a first peripheral unit operation, when said requested peripheral unit is already in operation.

19. A system according to Claim18, wherein:

said memory contains a device table having a section for each device in the system, where a device may be a peripheral unit, a controller means, or a bus, each device section containing a location for an address to a control block for that device when that device is in operation.

20. A system according to Claim 19 wherein:

each device section of said device table includes a location for storing an address to a control block for the first device operation to be requested and also a storage location to contain an address to the control block for the last device operation to be requested.

21. A system according to Claim 20 wherein:

a portion of said device table sections are set aside for input/output bus operations.

22. A system according to Claim 21 wherein a portion of said device table is set aside for controller means operations.

23. A system according to Claim 22 wherein a portion of said device table is set aside for peripheral unit operations.

24. A processing system having at least one central processor and a memory for storing a plurality of processes to be executed by said at least one central processor, which processes require different input/output events to occur, said system comprising:

an input/output processor means coupled to said memory;

a plurality of controller means;

a plurality of peripheral units; and

a plurality of buses coupling said peripheral units to various ones of said controller means and also coupling said controller means to said input/output processor means;

said input/output processor means including an input buffer to receive an input/output operation command specifying a particular unit and a reference to a control block in memory to control the operation of that unit, said input/output processor means also including á device table specifying the current status of activity of each of said units, said input/output processor means also including control means to address said memory to link the address therein of an input/output control block for a second request of a peripheral unit operation to the control block for a first peripheral unit operation, when said requested peripheral unit is already in operation.

25. A system according to Claim 24, wherein:

said memory contains a device table having a section for each device in the system, where a device may be a peripheral unit, a controller means, or a bus, each device section containing a location for an address to a control block for that device when that device is in operation.

26. A system according to Claim 25 wherein:

said data transfer means includes first means to transfer data segments between said disk cache area and said first portion of memory while said input/output processor means is transferring a track of data segments from one of said disk drive means to said disk cache area.

27. A system according to Claim 26 wherein:

said input/output means includes search means to search said disk cache area for a particular track when said at least one central processor calls for a READ operation from said first portion of said memory; and

said data transfer means includes second means to transfer said data segment to said designated track if that track resides in said disk cache area.

28. A system according to Claim 27 wherein:

said input/output processor means includes third means to transfer said data segment to a particular one of said tracks in a disk drive means when that segment is also stored in a corresponding track in said disk cache area.

29. A processing system having at least one central processor and a memory having a first portion for storing a plurality of processes to be executed by said at least one central processor, which processes require I/O operations to occur before their execution can be completed, a second portion of said memory being set aside as a disk cache area, said system comprising:

a plurality of disk mechanisms each having tracks for storing data and other information; and

input/output processor means coupled between said disk drive means and said memory for fetching an entire track of data for storage in said disk cache area of said memory when said at least one central processor requests a data segment to be fetched from a track in one of said disk drive means.

30. A system according to Claim 29 further including:

data transfer means coupled to said disk cache area of said memory and to said first portion of said memory, storing a plurality of processes, for transferring a data segment from said disk cache area to said first portion of memory when a process requires a data segment stored in said disk cache area.

31. A system according to Claim 30 wherein:

said data transfer means includes first means to transfer data segments between said disk cache area and said first portion of memory while said input/output processor means is transferring a track of data segments from one of said disk drive means to said disk cache area.

32. A system according to Claim 31 wherein:

said input/output means includes search means to search said disk cache area for a particular track when said at least one central processor calls for a READ operation from said first portion of said memory; and

said data transfer means includes second means to transfer said data segment to said designated track if that track resides in said disk cache area.

33. A system according to Claim 32 wherein:

said input/output processor means includes third means to transfer said data segment to a particular one of said tracks in a disk drive means when that segment is also stored in a corresponding track in said disk cache area.

34. A processing system having at least one central processor and a memory for storing a plurality of processes to be executed by said at least one central processor, which processes require different input/output operations to occur before their execution can be completed, said system comprising:

an input/output processing means;

a plurality of peripheral units;

a plurality of controller means for controlling said peripheral units; and

a plurality of buses coupling said peripheral units to said various ones of said controller means and said controller means to said input/output processing means;

said input/output processor means including an input buffer to receive an input/output operation command specifying a peripheral unit, a device table having an entry location for each peripheral unit in the system which table includes status information as to the availability of the peripheral unit and a path table means containing entries of various controllers that control a given peripheral unit and various buses that may be utilized to access said controller means and said peripheral unit;

said input/output processor means further containing control means coupled to said path table means to select a controller means and appropriate bus to access a given peripheral unit, which selection is dependent upon what combination of a controller means and a bus has the least input/output transfer load.

35. A system according to Claim 34 wherein:
a device entry in said device table specifies an initial entry to said path table means and the number of bus/controller means combinations that are available to access a particular peripheral unit.

36. A system according to Claim 35 wherein said input/output processor means further includes:
control means coupled to said device table and said path group means to sequence through the various combinations of controller means and buses specified by the peripheral unit entry in said device table.

37. A system according to Claim 36 further including:
a plurality of port adaptor means each coupling a plurality of said buses to said memory and to said input/output processor, each port adaptor means driving a number of said buses concurrently.

38. A system according to Claim 37 wherein:
said memory contains groups of control blocks for the various port adaptors, controller means and peripheral units, with respective control blocks being transferred to those port adaptors, controller means and peripheral units that have been selected.

39. In combination an input/output system and a memory, said system comprising:

an input/output processing means;

a plurality of peripheral units;

a plurality of controller means for controlling said peripheral units; and

a plurality of buses coupling said peripheral units to said various ones of said controller means and said controller means to said input/output processing means;

said input/output processor means including an input buffer to receive an input/output operation command specifying a peripheral unit, a device table having an entry location for each peripheral unit in the system which table includes status information as to the availability of the peripheral unit and a path table means containing entries of various controllers that control a given peripheral unit and various buses that may be utilized to access said controller means and said peripheral unit;

said input/output processor means further containing controller means coupled to said path table means to select a controller means and appropriate bus to access a given peripheral unit, which selection is dependent upon what combination of a controller means and a us has the least input/output transfer load.

40. A system according to Claim 39 wherein:

a device entry in said device table specifies an initial entry to said path table means and the number of bus/controller means combinations that are available to access a particular peripheral unit.

41. A system according to Claim 40 wherein said input/output processor means further includes:

control means coupled to said device table and said path group means to sequence through the various combinations of controller means and buses specified by the peripheral unit entry in said device table.

42. A system according to Claim 41 further including:

a plurality of port adaptor means each coupling a plurality of said buses to said memory and to said input/output processor, each port adaptor means driving a number of said buses concurrently.

43. A system according to Claim 42 wherein:

said memory contains groups of control blocks for the various port adaptors, controller means and peripheral units, with respective control blocks being transferred to those port adaptors, controller means and peripheral units that have been selected.

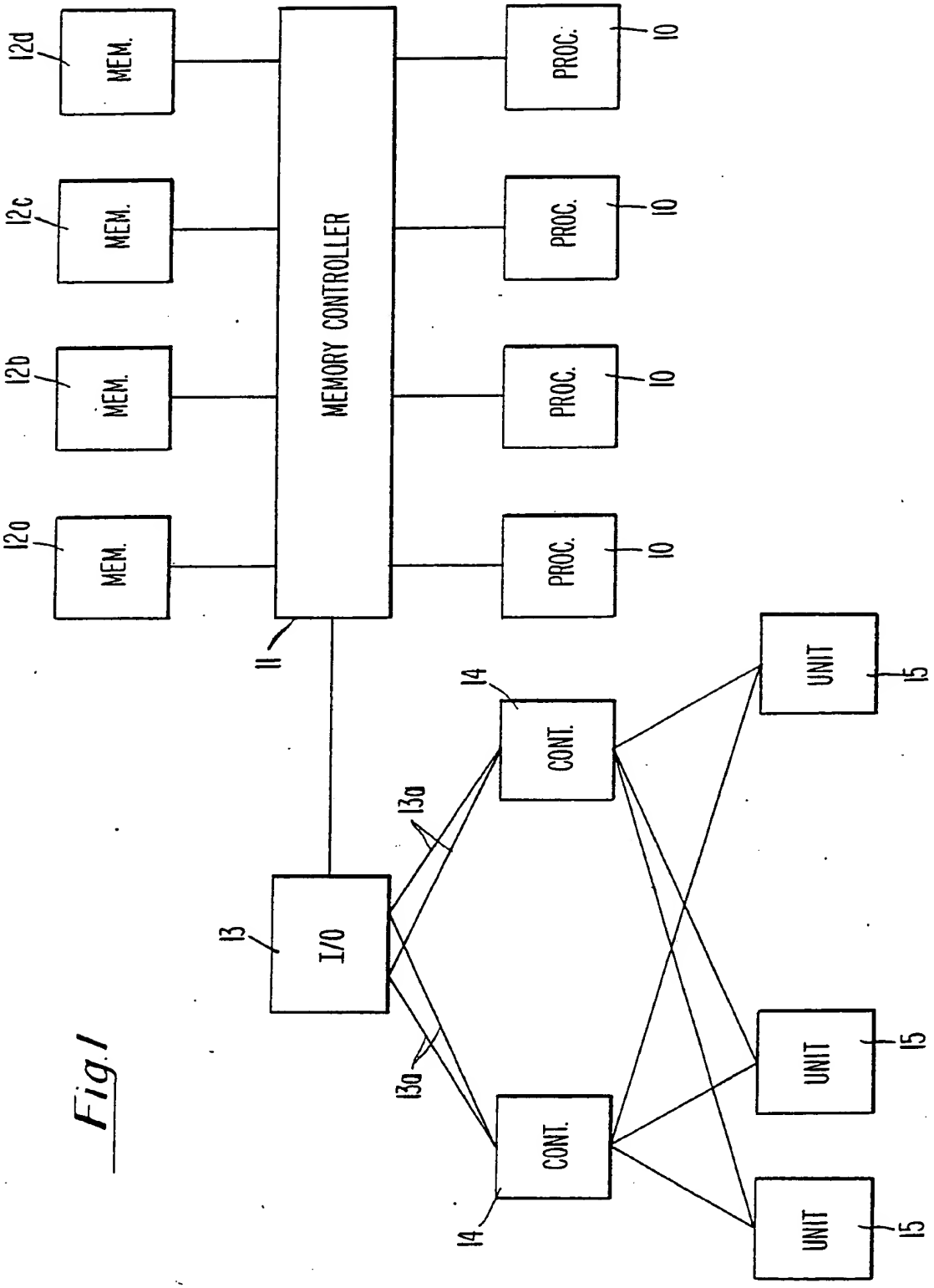

MEM.
12d
MEM.
12c
MEM.
12b
MEM.
12a
MEMORY CONTROLLER
11
PROC.
10
PROC.
10
PROC.
10
PROC.
10
I/O
13
13a
13a
CONT.
14
CONT.
14
UNIT
15
UNIT
15
UNIT
15

Fig.1

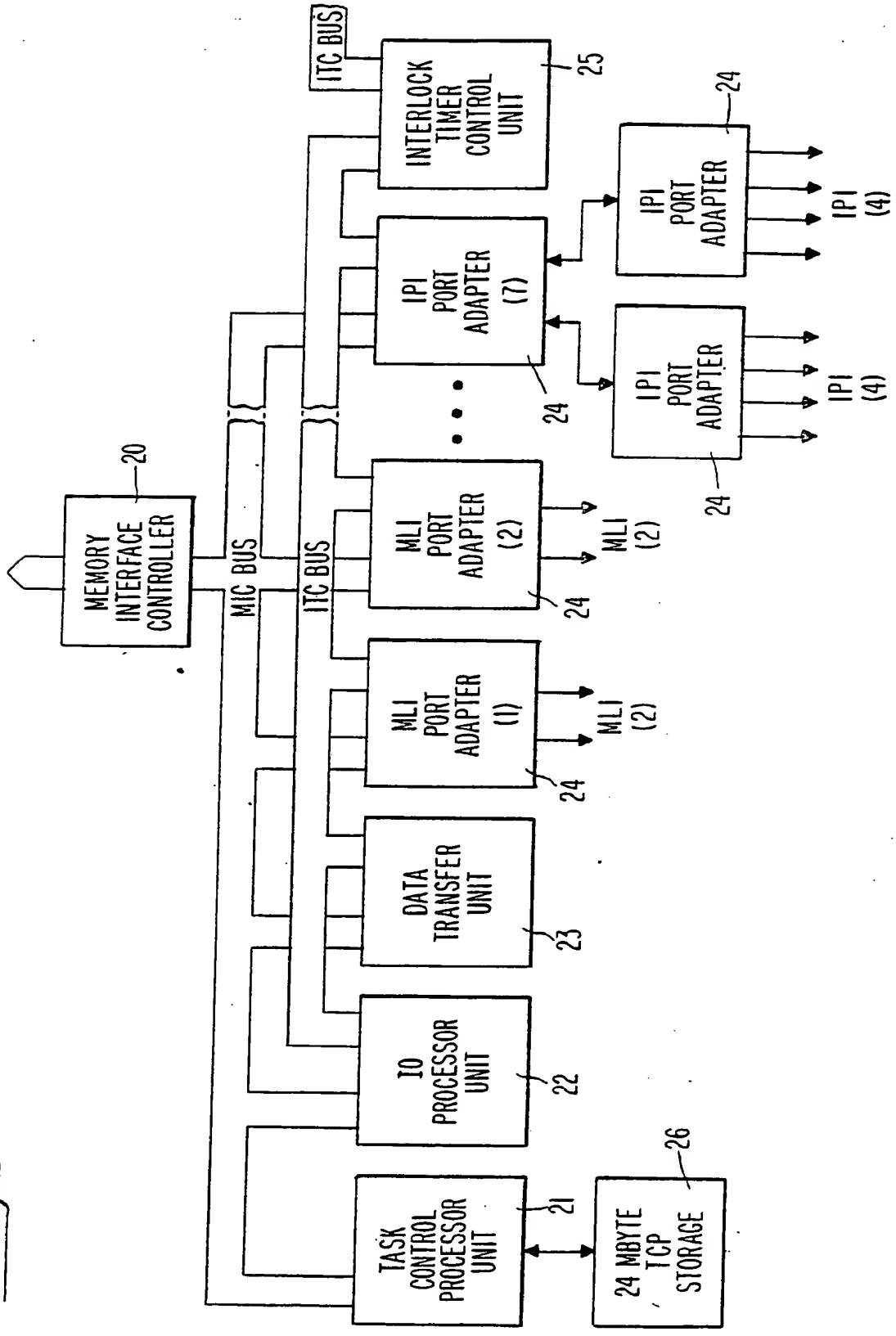

ITC BUS
INTERLOCK TIMER CONTROL UNIT
25
IPI PORT ADAPTER (7)
IPI PORT ADAPTER
IPI (4)
24
IPI PORT ADAPTER
IPI (4)
24
MEMORY INTERFACE CONTROLLER
20
MIC BUS
ITC BUS
MLI PORT ADAPTER (2)
MLI (2)
24
MLI PORT ADAPTER (1)
MLI (2)
24
DATA TRANSFER UNIT
23
IO PROCESSOR UNIT
22
TASK CONTROL PROCESSOR UNIT
21
24 MBYTE TCP STORAGE
26

Fig.2

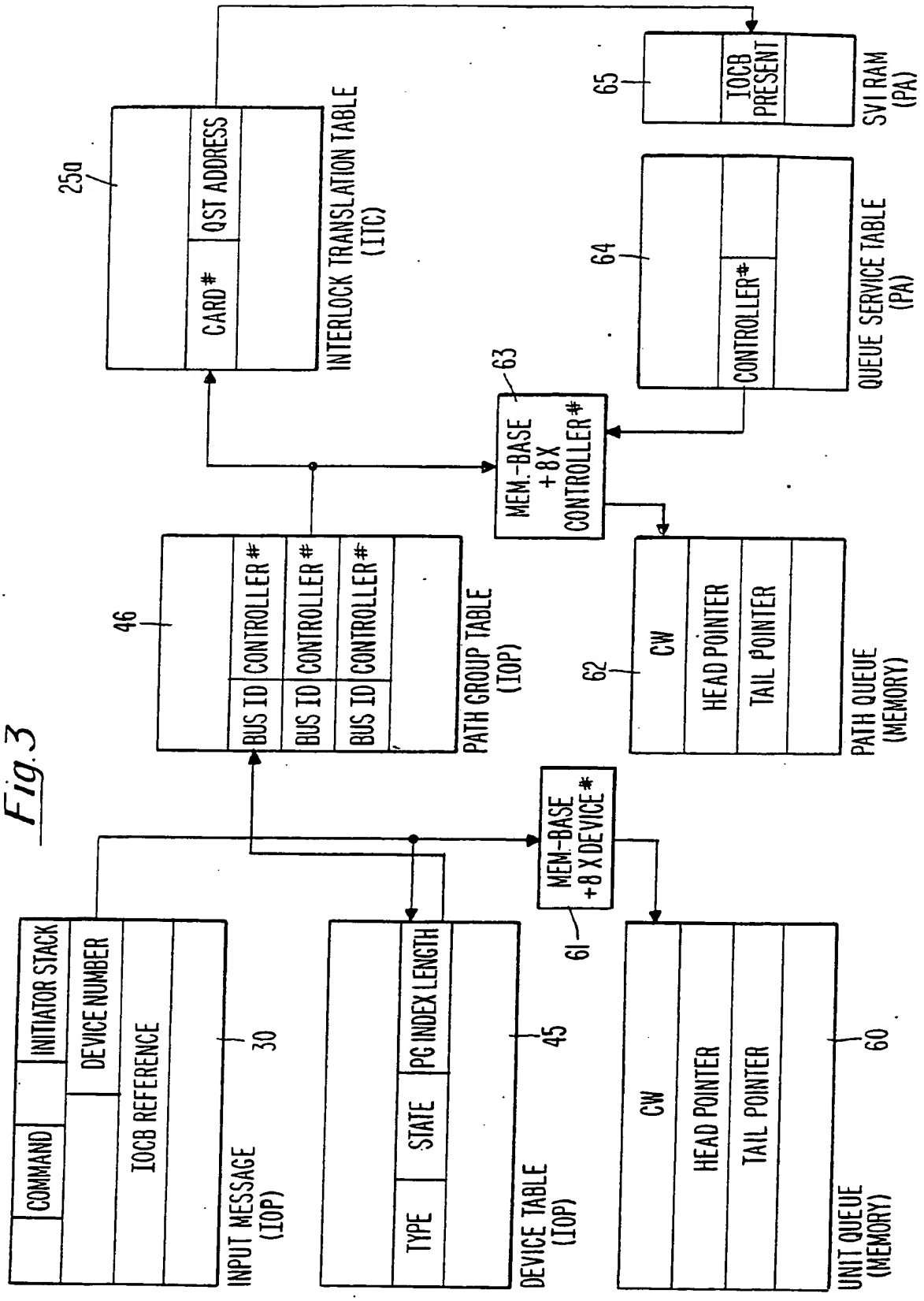

INPUT MESSAGE (IOP)
30
COMMAND
INITIATOR STACK
DEVICE NUMBER
IOCB REFFERENCE
DEVICE TABLE (IOP)
45
TYPE
STATE
PG INDEX LENGTH
UNIT QUEUE (MEMORY)
60
CW
HEAD POINTER
TAIL POINTER
MEM.-BASE +8 X DEVICE #
61
PATH GROUP TABLE (IOP)
46
BUS ID
CONTROLLER #
PATH QUEUE (MEMORY)
62
CW
HEAD POINTER
TAIL POINTER
MEM.-BASE +8 X CONTROLLER #
63
INTERLOCK TRANSLATION TABLE (ITC)
25a
CARD #
QST ADDRESS
QUEUE SERVICE TABLE (PA)
64
CONTROLLER #
SVI RAM (PA)
65
IOCB PRESENT

Fig. 3

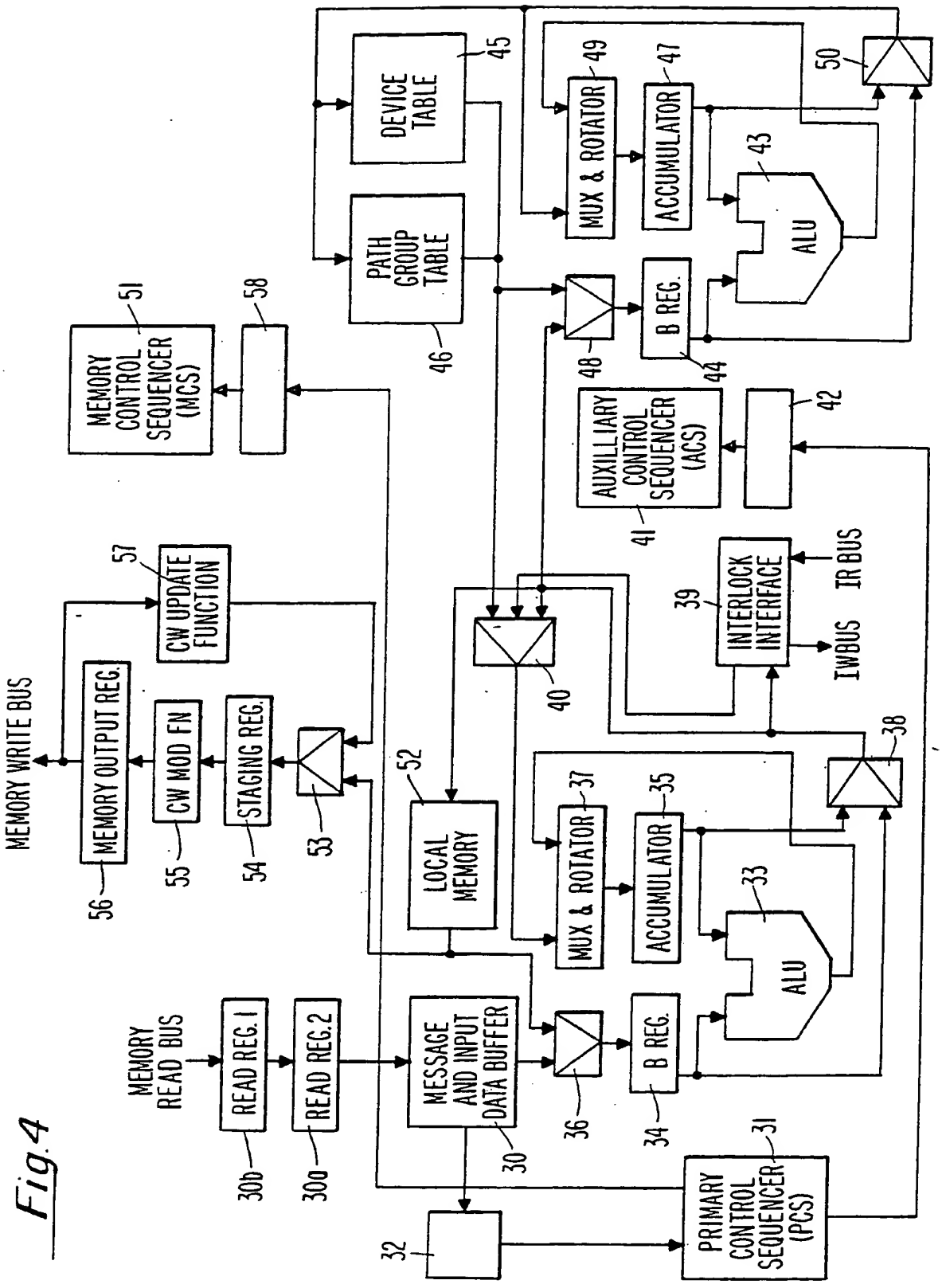

DEVICE TABLE
45
PATH GROUP TABLE
46
MUX & ROTATOR
49
ACCUMULATOR
47
ALU
43
50
B REG.
44
48
MEMORY CONTROL SEQUENCER (MCS)
51
58
AUXILLIARY CONTROL SEQUENCER (ACS)
41
42
INTERLOCK INTERFACE
39
IR BUS
IWBUS
40
CW UPDATE FUNCTION
57
MEMORY WRITE BUS
MEMORY OUTPUT REG.
56
CW MOD FN
55
STAGING REG.
54
53
LOCAL MEMORY
52
MUX & ROTATOR
37
ACCUMULATOR
35
ALU
33
38
MEMORY READ BUS
READ REG. 1
30b
READ REG.2
30a
MESSAGE AND INPUT DATA BUFFER
30
36
B REG.
34
PRIMARY CONTROL SEQUENCER (PCS)
31
32

Fig.4

Fig. 5A

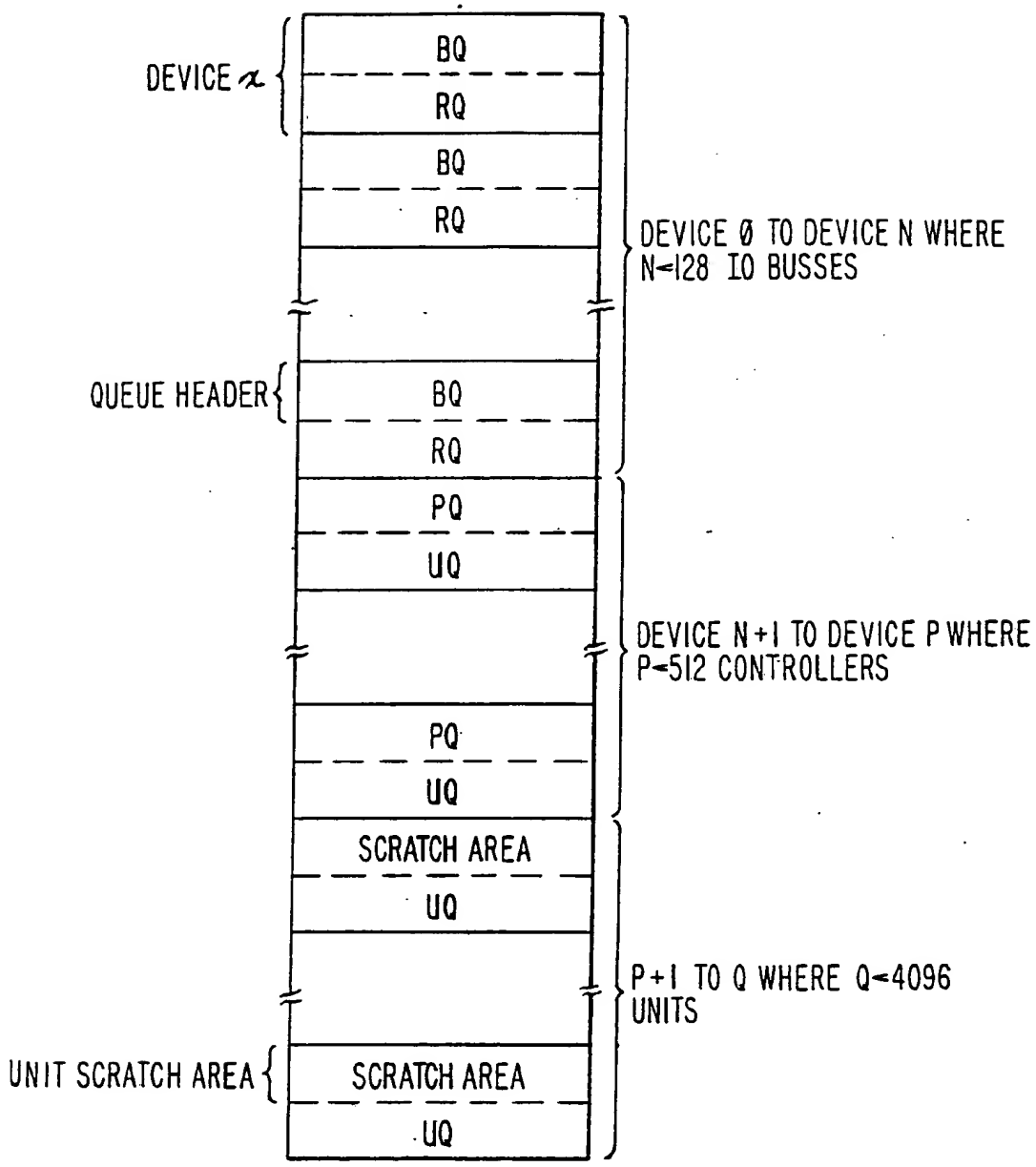

DEVICE x
BQ
RQ
BQ
RQ
DEVICE Ø TO DEVICE N WHERE N=128 IO BUSSES
QUEUE HEADER
BQ
RQ
PQ
UQ
DEVICE N+1 TO DEVICE P WHERE P=512 CONTROLLERS
PQ
UQ
SCRATCH AREA
UQ
P+1 TO Q WHERE Q=4096 UNITS
UNIT SCRATCH AREA
SCRATCH AREA
UQ

Fig. 5B

QUEUE HEADER

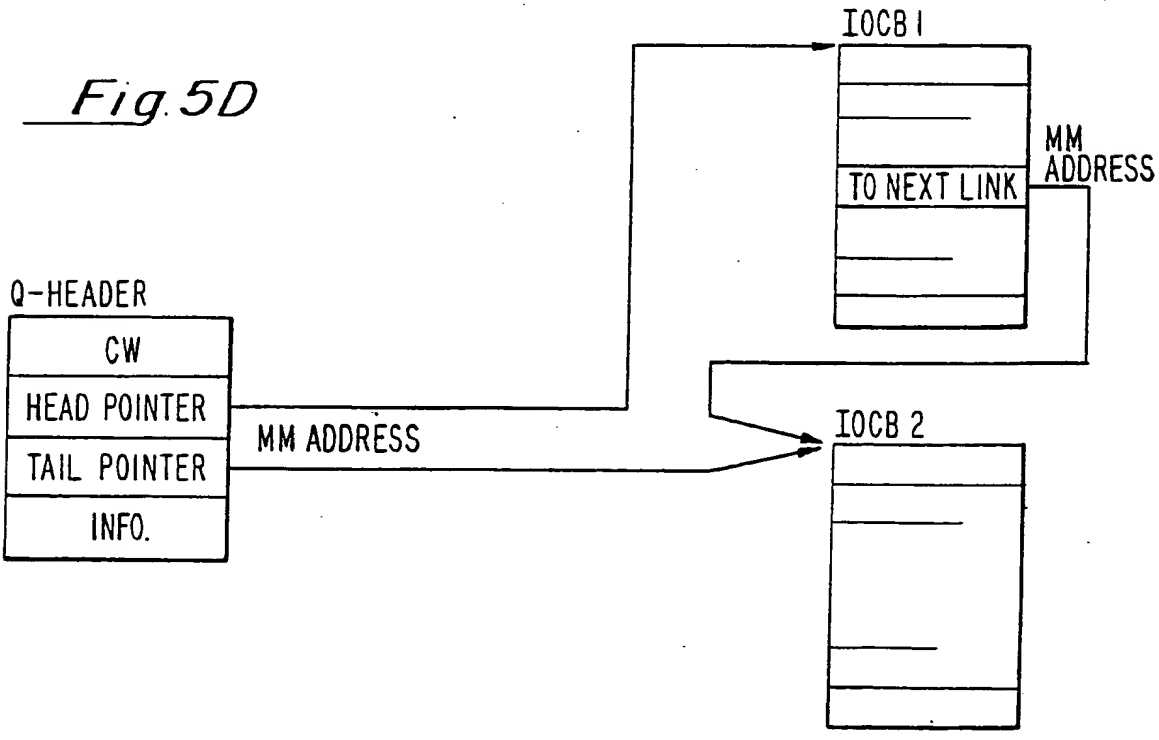
Q-CONTROL-WORD	CONTROL AND STATUS
HEAD-POINTER	POINTS TO FIRST IOCB
TAIL-POINTER	POINTS TO LAST IOCB
Q-INFO	ADDITIONAL INFORMATION

Fig. 5C

UNIT SCRATCH AREA

NEXT MIRROR ID	STARTING MIRROR SECTOR	MIRROR SET INFORMATION
PREVIOUS MIRROR ID	ENDING MIRROR SECTOR	MIRROR SET INFORMATION
WORK IOCB POINTER		WORK IOCB USED FOR DISK CACHING

Fig. 5D



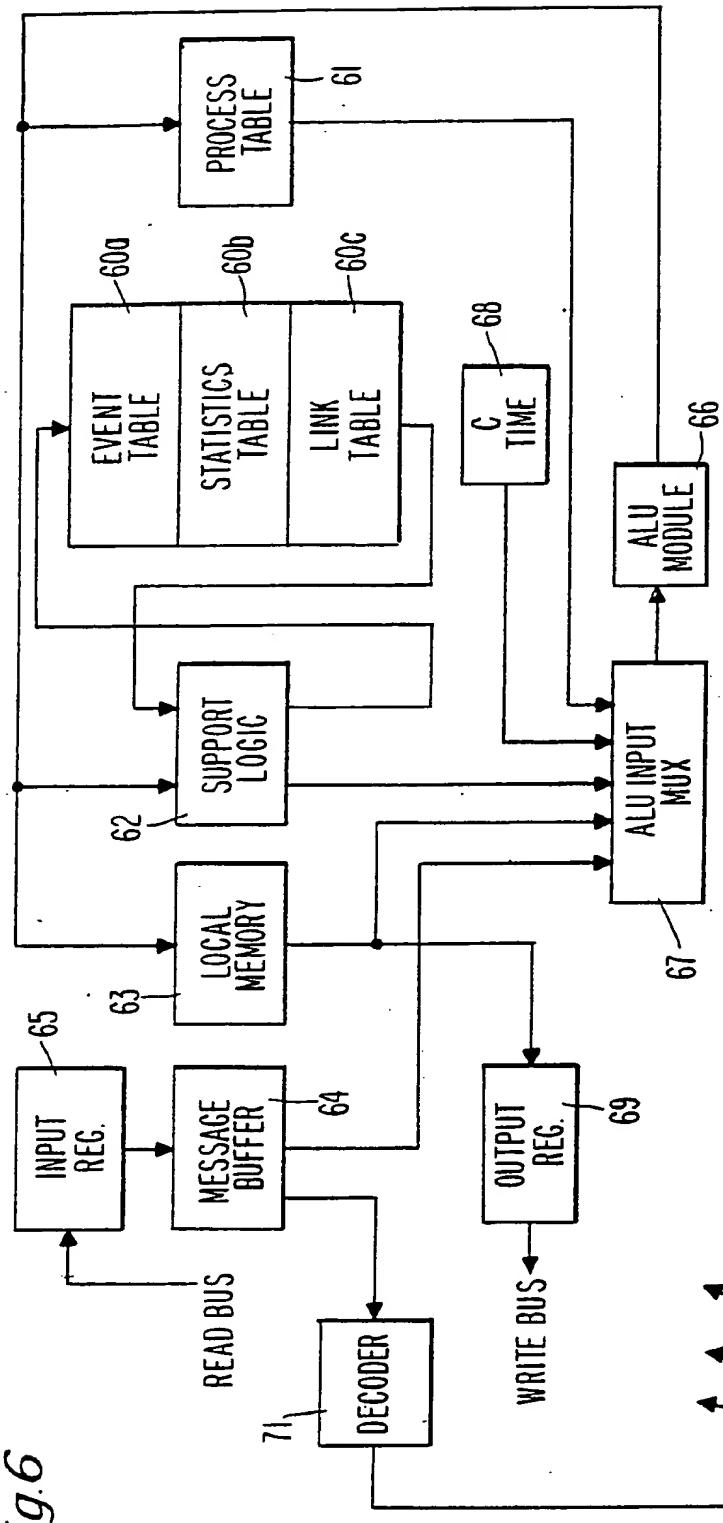

PROCESS TABLE
61
60a
60b
60c
EVENT TABLE
STATISTICS TABLE
LINK TABLE
68
C TIME
66
ALU MODULE
SUPPORT LOGIC
62
ALU INPUT MUX
LOCAL MEMORY
63
67
65
INPUT REG.
MESSAGE BUFFER
64
OUTPUT REG.
69
READ BUS
WRITE BUS
DECODER
71
70
SEQUENCE CONTROL STORE

Fig.6

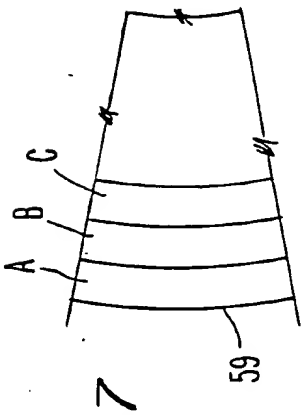

A
B
C
59

Fig.7

INTERNATIONAL SEARCH REPORT

International Application No PCT/US 87/02812

I. CLASSIFICATION OF SUBJECT MATTER (if several classification symbols apply, indicate all) *

According to International Patent Classification (IPC) or to both National Classification and IPC

IPC[4]: G 06 F 9/46; G 06 F 13/12

II. FIELDS SEARCHED

Minimum Documentation Searched 7

Classification System	Classification Symbols
IPC[4]	G 06 F 9/46; G 06 F 13/12; G 06 F 13/40

Documentation Searched other than Minimum Documentation to the Extent that such Documents are included in the Fields Searched *

III. DOCUMENTS CONSIDERED TO BE RELEVANT *

Category *	Citation of Document, 11 with indication, where appropriate, of the relevant passages 12	Relevant to Claim No. 13
X	EP, A, 0049521 (NAKANISHI) 14 April 1982 see page 3, lines 13-27; page 4, lines 10-15; page 7, lines 19-22; page 11, lines 20-28; page 12, lines 1-15; page 13, lines 1-24; figures 1,5,6 --	1
A	Communications of the A.C.M., volume 24, no. 10, October 1981, (New York, US), P.J. DENNING et al.: "Low contention semaphores and ready lists", pages 687-699 see the whole article --	1-11
A	IEEE Transactions on Computers, volume C-33, no. 7, July 1984, IEEE, (New York, US), R. Männer: "Hardware task/processor scheduling in a polyprocessor environment", pages 626-636 see page 627, left-hand column, lines 3-6; page 628, right-hand column,	1,3,4,8, 10,11

* Special categories of cited documents: 10

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier document but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art.

"&" document member of the same patent family

IV. CERTIFICATION

Date of the Actual Completion of the International Search	Date of Mailing of this International Search Report
26th February 1988	18 APR 1988
International Searching Authority	**Signature of Authorized Officer**
EUROPEAN PATENT OFFICE	P.C.G. VAN DER PUTTEN

Form PCT/ISA/210 (second sheet) (January 1985)

III. DOCUMENTS CONSIDERED TO BE RELEVANT (CONTINUED FROM THE SECOND SHEET)

Category *	Citation of Document, with indication, where appropriate, of the relevant passages	Relevant to Claim No.
	line 9 - page 630, left-hand column, line 24	
A	EP, A, 0064142 (HOFFMAN) 10 November 1982 see page 1, line 9 - page 2, line 11	1,3,4,8, 10,11
A	US, A, 4413317 (SWENSON) 1 November 1983 see column 2, lines 21-55; column 4, lines 43-48; column 5, lines 34-62; figures 1,2	12,18,20, 24-30,33
A	US, A, 4435755 (MERITT) 6 March 1984 see column 6, lines 35-43; column 7, lines 16-19	34,39
A	GB, A, 2020456 (LUIZ) 14 November 1979 see abstract; page 1, lines 74-130; page 2, lines 1-59; figure 2	34,39
P,X	WO, A, 87/02486 (JENNINGS) 23 April 1987 see the whole document cited in the application	1-11

ANNEX TO THE INTERNATIONAL SEARCH REPORT ON INTERNATIONAL PATENT APPLICATION NO. US 8702812 SA 19521

This annex lists the patent family members relating to the patent documents cited in the above-mentioned international search report. The members are as contained in the European Patent Office EDP file on 31/03/88
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
EP-A- 0049521	14-04-82	JP-A- 57064859	20-04-82
EP-A- 0064142	10-11-82	JP-A- 57187759	18-11-82
		US-A- 4394727	19-07-83
		CA-A- 1173971	04-09-84
US-A- 4413317	01-11-83	None	
US-A- 4435755	06-03-84	None	
GB-A- 2020456	14-11-79	NL-A- 7903614	12-11-79
		FR-A,B 2425676	07-12-79
		DE-A,B,C 2917441	15-11-79
		US-A- 4207609	10-06-80
		AU-A- 4564079	15-11-79
		JP-A- 54146941	16-11-79
		CA-A- 1116260	12-01-82
		AU-B- 521915	06-05-82
		CH-A- 637229	15-07-83
		SE-A- 7903939	09-11-79
		SE-B- 440960	26-08-85
WO-A- 8702486	23-04-87	EP-A- 0243402	04-11-87

EPO FORM P0179

For more details about this annex : see Official Journal of the European Patent Office, No. 12/82